Microsoft

Operating System

MIT Kerberos 5 (krb5 1.0) Interoperability

Technical Walkthrough

Abstract

This walkthrough examines the use of the MIT Kerberos interoperability features with the Microsoft® Windows® 2000 operating system.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

0599

CONTENTS

INTRODUCTION

The Microsoft® Windows® 2000 operating system implementation of the Kerberos version 5 protocol is designed for interoperability with other security services based on the MIT Kerberos version 5 reference implementation. The interoperability objectives for the Windows 2000 release support the following configurations:

- A Windows 2000 Server domain controller can serve as the Kerberos Key Distribution Center (KDC) server for MIT Kerberos-based client and host systems. UNIX systems can use **kinit** and the DES-CBC-MD5 or DES-CBC-CRC encryption type to authenticate to the Windows 2000 KDC.
- The Windows 2000 Kerberos Security Support Provider (SSP) implements the General Security Service Application Program Interface (GSS API) Kerberos Mechanism Token formats defined in RFC 1964. Windows 2000 does not provide the GSS API; instead, the Kerberos support is available to Microsoft Win32® applications using the SSPI APIs implemented by the Kerberos SSP. Client applications on UNIX using GSS API can obtain session tickets to services on Windows 2000, and can complete mutual authentication, message integrity, and confidentiality. (Context flags that are verified are GSS_C_MUTUAL_FLAG, GSS_C_REPLAY_FLAG, GSS_C_CONF_FLAG, GSS_C_INTEG_FLAG.)
- Windows 2000 Professional clients can be configured to use an MIT Kerberos server, with single sign on to the MIT KDC and a local Windows 2000 workstation account.

Interoperability with MIT Kerberos services requires minor configuration changes from the default installation. For example, a Windows 2000 workstation that uses an MIT-based Kerberos KDC server must be configured to locate the Kerberos realm and available KDC servers. Microsoft provides command-line tools to help with the configuration steps. These tools are:

- **Ksetup**. Configures alternate KDCs.
- **Ktpass**. Sets the password, account name mappings, and keytab generation for UNIX services that use the Windows 2000 Kerberos KDC.

The Kerberos configuration utilities are available on the Windows 2000 distribution media, in the *support**reskit**netmgmt**security* folder.

The Kerberos test utilities are available in the Windows 2000 software development kit (SDK), in the *mssdk**samples**winbase**security**win2000* folder. These tools are:

- **Gssclient**. Tests RFC 1964 interoperability (gss directory).
- **Gssserver**. Tests RFC 1964 interoperability (gss directory).
- **Klist**. Examines the Kerberos credential cache.

The Windows 2000 release has some known Kerberos interoperability limitations. These limitations include the following:

- Only DES-CBC-MD5 and DES-CBC-CRC encryption types are available for MIT interoperability.
- Hierarchical realm support for cross-platform trust between the Windows 2000 and MIT Kerberos realms is not included; however, transitive trust is supported between Windows 2000 domains in the domain tree.
- When a Windows 2000 system is in an MIT Kerberos realm, the user password change is not using the Kerberos Password Change protocol.
- Any upgraded user accounts and the administrator account in a new domain must have the password changed before non-Windows Kerberos clients can use them.

USING MIT KERBEROS CLIENTS

Creating Computer and User Accounts

Use the Active Directory™ Management tool to create computer and user accounts for the host and user security principals logging into the Windows 2000 Kerberos domain.

To configure the UNIX hosts

1. Use the Active Directory Management tool to create a new user account for the UNIX host:

   - Select the **Users** folder, right-click and select **New**, and then choose **user**.
   - Type the name of the UNIX host.

   The account can be in any container. It might be useful to create a new organizational unit (OU) for these accounts, and create them there.

2. Use **Ktpass** to create the keytab file and set up the account for the UNIX host, and then copy the keytab file to the UNIX system and merge the keytab file into /etc/krb5.keytab, as follows:

   - Use the following command to generate the UNIX host keytab file, map the principal to the account, and set the host principal password:

     ```
     C:> Ktpass –princ host/hostname@NT-DNS-REALM-NAME –mapuser
     account -pass password –out unixmachine.keytab
     ```

     where:
     – *hostname* is the host DNS name, for example, *foobar.microsoft.com*.
     – *NT-DNS-REALM-NAME* is the uppercase name of the Windows 2000 domain; for example, *NTDOM.MICROSOFT.COM*.
     – *account* is the user account for the computer.
     – *password* is a complex password for the account.

     Windows 2000 account names are not multipart as are Kerberos principal names. Because of this, it is not possible to directly create an account of the name *host/hostname.dns.com*. Such a principal instance is created through service principal name mappings. In this case, an account is created with a meaningful name *hostname* and a service principal name mapping is added for *host/hostname.dns.com*. This is the purpose of using **Ktpass** with the **–princ** and **–mapuser** switches.

   - Securely transfer the keytab file (unixmachine.keytab from the example above) to the UNIX host. Then, merge the keytab file with the keytab file for the UNIX computer. The UNIX commands to merge the keytab file are:

     ```
     % ktutil
     ktutil: rkt unixmachine.keytab
     ktutil: list
     ```

     The output should appear similar to the following:

```
slot KVNO Principal
---- ---- ---------
   1    1 host/foobar.microsoft.com@NTDOM.MICROSOFT.COM

ktutil: wkt /etc/krb5.keytab
ktutil: q
```

- Edit the file (/etc/krb5.conf) to refer to the Windows 2000 domain controller as the Kerberos KDC. The krb5.conf file entries should be similar to the following:

```
[libdefaults]
default_realm  = NTDOM.MICROSOFT.COM
default_tkt_enctypes = des-cbc-md5 ; or des-cbc-crc
default_tgs_enctypes = des-cbc-md5   ; or des-cbc-crc

[realms]
NTDOM.MICROSOFT.COM = {
kdc = server3.ntdom.microsoft.com:88
}
```

- The default encryption type entries, default_txx_enctype, are optional. However, if the MIT client receives an encryption type error, set the default encryption type to either DES-CBC-MD5 or DES-CBC-CRC.
- If your UNIX computer's DNS name does not include the realm name—if for example, *foobar.microsoft.com* does not include *ntdom.microsoft.com*—you may be required to map the hostname to the Kerberos realm name manually, as follows:

```
[domain_realm]
.foobar.microsoft.com  = NTDOM.MICROSOFT.COM
```

  Without this entry, your Kerberos applications might try to connect to the wrong realm and fail, which can be frustrating to debug.

  See the Kerberos version 5 manual pages for more information on *krb5.conf*.

- Be sure that your system clocks are synchronized (within two minutes) to the KDC system's clock. Otherwise, Kerberos authentication will fail due to clock skew errors.

3. Finally, you need to create UNIX accounts that correspond to the Windows 2000 domain accounts so that the login process will know to use Kerberos authentication. You can do this by using the **vipw** command or other administration tools, depending on how you manage UNIX accounts. See the Kerberos version 5 manual pages for more information.

SUPPORT FOR MIT KERBEROS SERVICES

Services running on UNIX systems can be configured with service instance accounts in the Active Directory. This allows full interoperability—MIT Kerberos clients and servers on UNIX systems can authenticate using the Windows 2000 Kerberos server, and Windows 2000 clients can authenticate to Kerberos services that support GSS API.

Unlike Kerberos principal names, Windows 2000 account names are not multipart. Because of this, it is not possible to directly create an account of the name *sample/unix1.ntdom.microsoft.com*. Such a principal instance is created through the service principal name mappings.

To create a service instance account in the Active Directory

1. Use the Active Directory Management tool to create a user account for the UNIX service; for example, create an account with the name *sampleUnix1*.
2. Use the **Ktpass** tool to set up an identity mapping for the user account. Use this command:

   ```
   C:> Ktpass –princ service-instance@REALM
   –mapuser account-name -pass password
   -out unixmachine.keytab
   ```

   The format of the Kerberos service-instance name is: *service**/**host**.**realm_name*, for example:

   ```
   C:> ktpass –princ
   sample/unix1.ntdom.microsoft.com@NTDOM.MICROSOFT.COM -mapuser
   sampleUnix1 –pass password –out unix1.keytab
   ```

   In this case, an account is created with a meaningful name *sampleUnix1*, and a service principal name mapping is added for *sample/unix1.ntdom.microsoft.com*. This is the purpose of using **Ktpass** with the –**princ** and –**mapuser** switches.
3. Merge the keytab file with the /etc/krb5.keytab file on the UNIX host.

Note You cannot map multiple service instances to the same user account.

USING AN MIT KDC WITH A WINDOWS 2000 WORKSTATION

For the Windows 2000 workstation to use an MIT Kerberos KDC, you must configure both the UNIX KDC server and the workstation as described next.

To configure the UNIX KDC server and the Windows 2000 workstation

1. Run the **Ksetup** utility to configure the UNIX KDC server and realm for the workstation to use (for details, see the **Ksetup** section later in this document):

   - In the MIT realm, create a host principal for the computer. Use the command:

     ```
     Kadmin –q "ank –pw password host/machine-name.dns-domain_name"
     ```

     For example, if the Windows 2000 workstation name is *W2KW* and the UNIX realm name is *ATHENA.MIT.EDU*, the principal name is *host/w2kw.athena.mit.edu*.

   - Since an MIT Kerberos realm is not a Windows 2000 domain, the computer must be configured as a member of a workgroup. This is automatically taken care of when you set the Kerberos realm and add a KDC server as follows:

     ```
     C:> Ksetup /setdomain MIT.MICROSOFT.COM
     C:> Ksetup /addkdc MIT.MICROSOFT.COM mitkdc.microsoft.com
     ```

   - Set the local machine account password, as follows:

     ```
     C:> Ksetup /setmachpassword password
     ```

2. Restart your computer for the changes to take effect. (This is a required step.) Whenever changes are made to the external KDC configuration, a restart is required.

3. Use **Ksetup** to configure single sign on to local workstation accounts. Define the account mappings; this will map local machine accounts to Kerberos principals. For example:

   ```
   C:> Ksetup /mapuser auser@MIT.MICROSOFT.COM guest
   C:> Ksetup /mapuser * *
   ```

   Note that the second command maps clients to local accounts of the same name.

4. Use **Ksetup** with no arguments to see the current settings. (Note that the KDC server[s] is not shown.)

SETTING TRUST WITH AN MIT KERBEROS REALM

You can set up a trust relationship between Windows 2000 domains and MIT Kerberos domains. The following procedure sets up trust between Windows 2000 domain *MITCOMPAT.NTTEST.MICROSOFT.COM* and MIT Kerberos realm *MIT.MICROSOFT.COM*.

To set up access to services

Workstation computers that use services in an MIT realm need to have a realm entry added. To do this, use the **Ksetup** command on each system that uses the MIT realm for services.

```
C:> Ksetup /addkdc MIT.MICROSOFT.COM mitkdc.microsoft.com
```

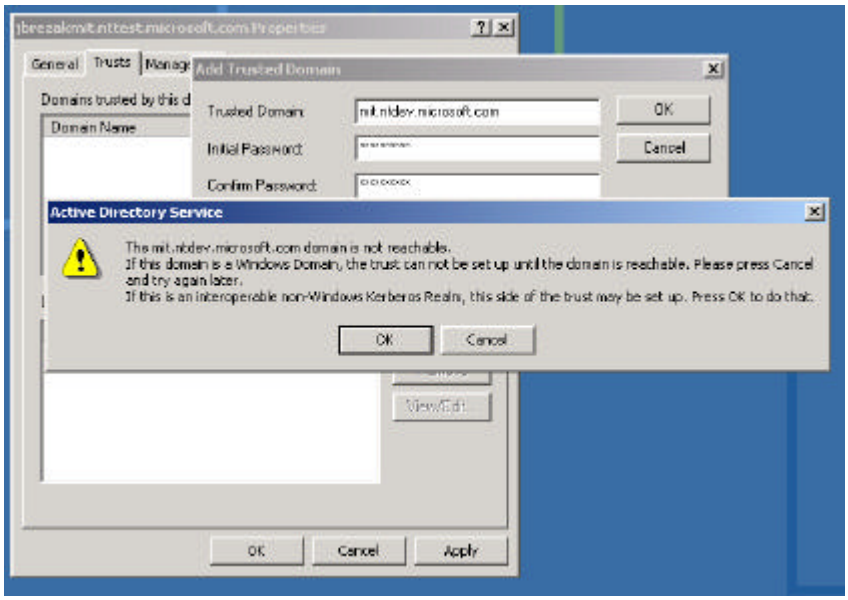
These mappings are stored in the registry under *HKLM\System\CurrentControlSet\Control\LSA\Kerberos\Domains*. To deploy realm configuration data to multiple computers, use the group policy mechanism instead of using **Ksetup** explicitly on individual computers.

To set up trust

1. On the domain controller for the Windows 2000 domain, use the following command to set up the configuration for the foreign MIT realm:

   ```
   C:> Ksetup /addkdc MIT.MICROSOFT.COM mitkdc.microsoft.com
   ```

2. Start the Domain Tree Management tool. Click **Programs**, then **Administrative tools**, and then **Active Directory Domains and Trusts**.
3. Right-click on **Properties** of your domain, then select the **Trust** tab and press **Add**. The passwords used in this step are used in Step 5.
4. Create a trusted domain relationship with the MIT Kerberos realm. When prompted if this is a non-Windows Kerberos Realm, click **OK**.



5. Use the following commands to create cross-realm principals in the foreign MIT realm:

```
C:> Kadmin –q “ank –pw password
krbtgt/MITCOMPAT.NTTEST.MICROSOFT.COM@MIT.MICROSOFT.COM”

C:> Kadmin –q “ank –pw password
krbtgt/MIT.MICROSOFT.COM@MITCOMPAT.NTTEST.MICROSOFT.COM”
```

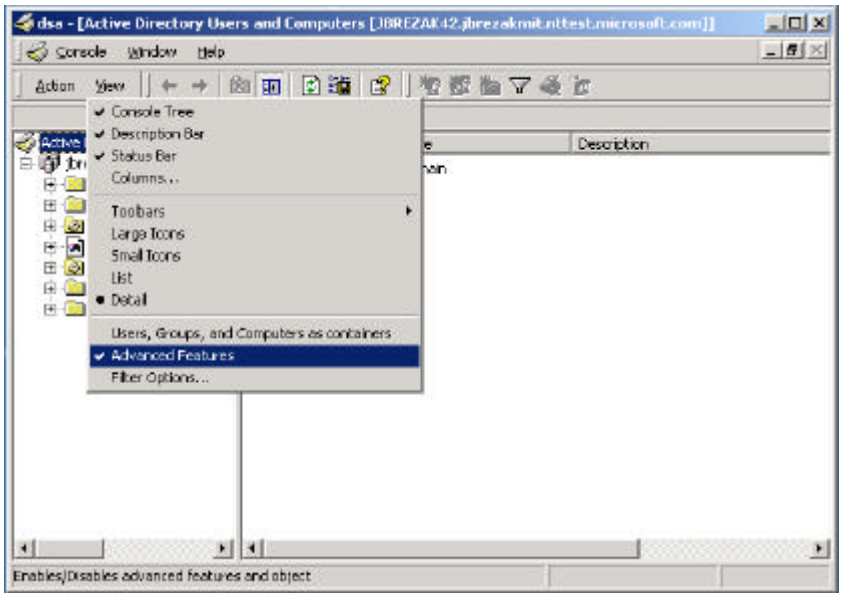
Instead of using the Domain Tree Management user interface (UI), you can use the Netdom tool to establish trust to a MIT Kerberos realm. The tool is located in the *\\support\reskit\netmgmt* folder on the distribution media. See the tool Help menu for details.

Creating Account Mappings

Account mappings are used to map a foreign Kerberos identity (in a trusted MIT Kerberos realm) to a local account identity in the domain. These account mappings are managed through the Active Directory Management tool.

To create a mapping

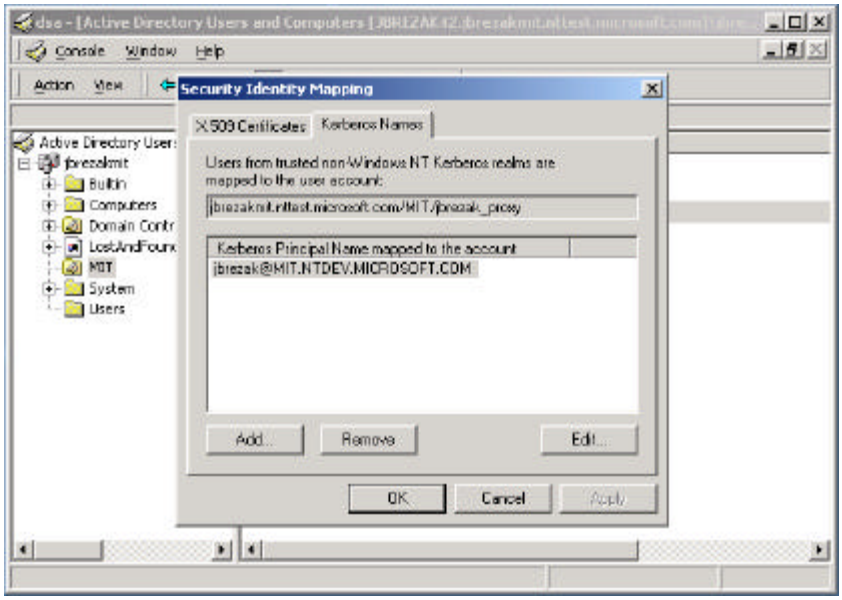
1. Start the **Directory Management** tool. Point to **Programs**, then **Administrative tools**, and then Active **Directory Users and Computers**.
2. Start advanced features by clicking **View**, and then **Advanced Features.**



3. Locate the account to which you want to create mappings, and right-click to view **Name Mappings**. This example uses the account *jbrezak_proxy*.

4. Click the **Kerberos Names** mappings tab.

5. Add a principal from the foreign MIT realm. This example uses jbrezak@MIT.NTDEV.MICROSOFT.COM.



USING MIT SAMPLE APPLICATIONS

The MIT Kerberos Krb5-1.0 distribution media comes with sample applications that demonstrate Kerberos using both the Krb5 and GSS APIs. These sample applications run properly on a UNIX system configured to use the Windows 2000 KDC. The samples are located in the following directories:

- /krb5-1.0.5/src/appl/sample
- /krb5/krb5-1.0.5/src/appl/gss-sample

Configure the UNIX host to use the Windows 2000 KDC, create a user account in the Active Directory, and set the password on the account. Verify that you can use **kinit** to authenticate from the UNIX host to the Windows 2000 KDC.

Sample Kerberos Client and Server Application

You can run the sample Kerberos client and server application using the Windows 2000 KDC.

To run the sample Kerberos application

1. Configure the sserver to start from inetd by making an entry in //etc/services. See the Kerberos version 5 manual page for *sserver* for more information.
2. Create a service instance account in the Active Directory. Follow the steps described in the previous section, “Support for MIT Kerberos Services.” To summarize, the steps are:
   - Use the Active Directory Management tool to create a **user** account for the UNIX service and name it *sample*.
   - Use the **Ktpass** tool to set up an identity mapping for the host account:

     ```
     C:> ktpass –princ service-instance
     –mapuser account-name -pass password
     -out unixmachine.keytab
     ```

     The format of the Kerberos service-instance name is: *service/host.realm name*., for example:

     ```
     sample/unix1.ntdom.microsoft.com
     ```

   - Merge the keytab file with the /etc/krb5.keytab file on the UNIX host.
3. On the UNIX host, use **kinit** on your user account and use **klist** to verify that you have a ticket to the krbtgt/DOMAIN.NAME@REALM.NAME principal.
4. Run the **sclient** program. Assuming the UNIX host name is pdcunix, the command line (using default values for the [port] and [service] arguments) is:

   ```
   % sclient pdcunix
   sendauth succeeded, reply is:
   reply len 44, contents:
   You are user@NTDOM.MICROSOFT.COM
   ```

GSS Sample Client and Server Applications

You can also run the GSS sample application, GSS server, and GSS client using Windows 2000 KDC as the authentication server. As with the Kerberos sample client/server, you can use *sample* as the service name. See the readme file in the gss-sample directory for more information.

To run the GSS sample client and server application using the Windows 2000 KDC

1. Use the Windows 2000 *sample* user account created for the Kerberos sample as described above.
2. Start the GSS server. Text similar to the following will be displayed.

```
% gss-server sample &
Start the GSS client:
% gss-client <host> sample “Message”

The following example output shows the result of running the GSS
client and server application on UNIX with the Windows NT KDC. The
user is authenticated as fred, the host is pdcunix, and the domain is
PETEBRTDOM.NTDEV.MICROSOFT.COM.
% gss-client pdcunix sample “Kerberos interop works great!”
"fred@PETEBRTDOM.NTDEV.MICROSOFT.COM" to
"sample/pdcunix.petebrtdom.ntdev.microsoft.com@PETEBRTDOM.NTDEV.MICRO
SOFT.COM", lifetime 34705, flags 36, locally initiated, open
Name type of source name is { 1 2 840 113554 1 2 2 1 }.
Mechanism { 1 2 840 113554 1 2 2 } supports 6 names
  0: { 1 2 840 113554 1 2 1 1 }
  1: { 1 2 840 113554 1 2 1 2 }
  2: { 1 2 840 113554 1 2 1 3 }
  3: { 1 2 840 113554 1 2 1 4 }
  4: { 1 2 840 113554 1 2 2 1 }
  5: { 1 2 840 113554 1 2 2 2 }
Sending init_sec_context token (size=534)...continue needed...

context flag: GSS_C_MUTUAL_FLAG
context flag: GSS_C_REPLAY_FLAG
context flag: GSS_C_CONF_FLAG
context flag: GSS_C_INTEG_FLAG
Signature verified.
```

The MIT Kerberos Krb5-1.0 release of the GSS samples uses reverse name lookups to identify the target server principal name. Therefore, the principal name for the GSS service must be in the form *service/host.domain.name*, where the domain name must be the same as the Windows 2000 domain name. If your UNIX host DNS domain name is different from the Windows 2000 domain name (they can be the same or different DNS domains), you need to define a DNS name entry for the host.

For example, if the DNS host name is *unix1.test.foobar.org*, the Windows 2000 domain name is *NTDOM.TEST.FOOBAR.ORG*, and the GSS server principal name is *sample/unix1.ntdom.test.foobar.org*, you would create the following entry in the /etc/hosts file :

```
x.x.x.x  unix1.ntdom.test.foobar.org  unix1.test.foobar.org  unix1
```

where *x.x.x.x* is your IP address.

Note If the host DNS domain name is different from the Windows 2000 domain DNS name, you might see a GSS client error verifying the data integrity.

Testing GSS API (RFC-1964) Interoperability

You can demonstrate the interoperability between the Windows 2000 Kerberos Security Support Provider and the GSS-Krb5 mechanism by running the Krb5 GSS sample applications on UNIX and a port of the GSS sample applications on Windows 2000.

The source code for the sample applications for the GSS client, and for the GSS server port to Windows 2000 using SSPI are available in the samples area of the Microsoft Windows 2000 SDK CD-ROM. The calls to GSS API were replaced with equivalent calls to SSPI, the corresponding Win32 network security interface.

This example uses the Windows 2000 domain controller as the KDC.

To run the GSS server application using the Windows 2000 KDC

1. Build the sample application GSS server in the platform SDK.
2. Use the *sample* account from the sclient example.
3. Run the SSPI-version of GSS server with the following example command:

   ```
   c:> gss-server sample/petebrtdc.ntdev.microsoft.com foobar
   PETEBRTDOM.NTDEV.MICROSOFT.COM
   ```

4. On the UNIX system, start the GSS client, and specify the Windows 2000 host and service name. Note that the service name is in the form *service@Windows NT host DNS name*. An example of the command and the subsequent output (from the sample) follows.

```
% gss-client petebrtdc sample@petebrtdc.ntdev.microsoft.com
            “Kerberos interop works great!”
The output of gssserver running on Windows NT looks similar to the
following:
D:\> gssserver
Usage: gss-server [-port port] [-verbose]
       [-inetd] [-logfile file] [service_name] [service_password]
[service_realm]

D:\> gssserver sample/petebrtdc.ntdev.microsoft.com foobar
PETEBRTDOM.
NTDEV.MICROSOFT.COM
context flag: GSS_C_MUTUAL_FLAG
context flag: GSS_C_REPLAY_FLAG
context flag: GSS_C_CONF_FLAG
Accepted connection: "PETEBRTDOM.NTDEV.MICROSOFT.COM\fred"
Received message: "Kerberos interop works great!"
```

You can use **Klist** to verify the tickets issued by the Windows 2000 KDC for the sample server running on the UNIX system and the Windows 2000 system. The output will be similar to the following:

```
% klist
Ticket cache: /tmp/krb5cc_ttyv0
Default principal: fred@PETEBRTDOM.NTDEV.MICROSOFT.COM

Valid starting      Expires             Service principal
12 Sep 97 16:19:49  13 Sep 97 02:18:48
krbtgt/PETEBRTDOM.NTDEV.MICROSOFT.COM@PETEBRTDOM.NTDEV.MICROSOFT.COM
12 Sep 97 16:19:56  13 Sep 97 02:18:48
sample/pdcunix.petebrtdom.ntdev.microsoft.com@PETEBRTDOM.NTDEV.MICROS
OFT.COM
12 Sep 97 17:07:23  13 Sep 97 02:18:48
sample/petebrtdc.ntdev.microsoft.com@PETEBRTDOM.NTDEV.MICROSOFT.COM
```

To run the GSS server application using the Windows 2000 KDC

1. Build the sample application *gssclient* in the platform SDK.
2. Use the *sample* account from the *sclient* example.
3. Run the MIT version of GSS server with the following example command:

   ```
   $ gss-server sample@mit.nttest.Microsoft.com
   ```

4. On the Windows 2000 system, start gssclient.exe to connect with the MIT server.

```
C:> gssclient mithost sample/mit.nttest.microsoft.com Hello
LoadLib = 77583214
checking mithost...calling host mithost, name
sample/mit.nttest.Microsoft.com@MIT.NTDEV.MICROSOFT.COM, msg "Hello".
Sending init_sec_context token (size=2055)...continue needed...

Block Size is 8, inbuffer = 6
Signature verified.

C:> klist tickets
Cached Tickets:
        Server: krbtgt/PETEBRDOM.NTDEV.MICROSOFT.COM@
PETEBRTDOM.NTDEV.MICROSOFT.COM
        KerbTicket Encryption Type: KERB_ETYPE_DES_CBC_MD5
                Start Time: 2/22/1999 10:43:49
                End Time: 3/24/1999 10:43:49
                Renew Time: 3/24/1999 10:43:49
                EncryptionType: 3
                TicketFlags: (0x40c00000) forwardable renewable
initial
        Server:
sample/mit.nttest.Microsoft.com@PETEBRDOM.NTDEV.MICROSOFT.COM
        KerbTicket Encryption Type: KERB_ETYPE_DES_CBC_MD5
                Start Time: 2/22/1999 10:41:18
                End Time: 2/22/1999 20:41:18
                Renew Time: 3/1/1999 10:41:18
                EncryptionType: 3
                TicketFlags: (0x40800000) forwardable renewable
```

FOR MORE INFORMATION

For the latest information on Windows 2000, visit our World Wide Web site at http://www.microsoft.com/ntserver/.

For the latest information on the Windows 2000 Beta 3, visit the World Wide Web site at http://www.microsoft.com/windows/server/.

Before You Call for Support

Please keep in mind that Microsoft does not support these walkthroughs. The purpose of the walkthroughs is to facilitate your initial evaluation of the Microsoft Windows 2000 features. For this reason, Microsoft cannot respond to questions you might have regarding specific steps and instructions.

Reporting Problems

Problems with Microsoft Windows 2000 Beta 3 should be reported by way of the appropriate bug reporting channel and alias. Please make sure to adequately describe the problem so that the testers and developers can reproduce it and fix it. Refer to the Release Notes included on the Windows 2000 Beta 3 distribution media for some of the known issues.